

# Safeguarding Privacy by Reliable Automatic Blurring of Faces in Mobile Mapping Images

Steven Puttemans, Stef Van Wolputte and Toon Goedemé

*EAVISE Research Group, KU Leuven, Jan Pieter De Nayerlaan 5, Sint-Katelijne-Waver, Belgium.  
[steven.puttemans, toon.goedeme]@kuleuven.be, stefvanwolputte@gmail.com*

**Keywords:** Cycloramic Imagery, Mobile Mapping, Pedestrian Detection, Application Specific Constraints, Soft Blurring.

**Abstract:** When capturing images in the wild containing pedestrians, privacy issues remain a major concern for industrial applications. Our application, collecting cycloramic mobile mapping data in crowded environments, is an example of this. If the data is processed and accessed by third parties, privacy of pedestrians must be ensured. This is where pedestrian detectors come into play, used to detect individuals and privacy mask them through blurring. The problem of undesired false positive detections, typical for pedestrian detectors and unavoidable, still leaves undesired areas of the images being blurred. We tackled this problem using application-specific scene constraints, modelled by a height-position mapping based on scene-specific pedestrian annotation data, combined with reducing the field of interest and case-specific false positive elimination classifiers. We applied a soft blurring technique to avoid the artificial look of simply applying Gaussian blurring to the found detections, which results in an effective fully-automated masking pipeline for privacy safeguarding in mobile mapping images. We prove that we can use pre-trained pedestrian detection models, but by collecting a limited amount of application-specific annotations and by exploiting scene-specific constraints, we are able to boost the detection accuracy enormously.

## 1 INTRODUCTION

In mobile mapping applications, a vehicle equipped with cameras is used to grab images in order to give the user a digital view of the surroundings. This is repeated at preset intervals in order to ensure that the complete surroundings of the car are being captured. Companies like Google, but also local land surveying offices, are carrying out such measuring campaigns to make digital images of streets across the globe. When collecting all this data, one can imagine that the amount of data increases drastically once someone is capturing larger projects, e.g. the ‘Google Street View’ application. The goal of capturing all this data is providing users with fast, accurate and detailed data measurements for producing all kinds of 2D and 3D geographical information systems.

Avoiding pedestrians walking around when capturing mobile data is nearly impossible, which raises the question of privacy issues when they are. Especially when this data is shared with or sold to industrial partners, it is important that the privacy of these pedestrians is guaranteed. Therefore companies are continuously looking for robust solutions able to filter out privacy-sensitive content from the captured data.

One solution could be to manually browse the data, indicating every pedestrian and making them privacy-safe by applying a blurring filter to the annotations. In the case that the amount of data is rather limited, this might be the fastest and most accurate solution. However when the data size rises over several millions of captured images a week, one immediately notices that this approach is no longer suitable. In those cases an automated unsupervised approach is preferred. One of the most frequently used techniques in tackling this problem is applying pedestrian detection algorithms like (Dalal and Triggs, 2005; Viola and Jones, 2001; Dollár et al., 2009; Felzenszwalb et al., 2008) on the captured mobile mapping data, marking possible pedestrian-like areas in the image. These in turn can then be blurred or cut out, to avoid transferring privacy-sensitive data.

A major downside of existing pedestrian detectors is that they require the manual selection of a threshold on the detection certainty score to find a good balance between finding actual pedestrians in the image and avoiding false positive detections. If the threshold is set too strict, we will only detect pedestrians but be unable to find all of them, and thus privacy issues arise again. If we put the threshold too sloppy, all

pedestrians will be found, but similar objects or areas will trigger a false positive detection such that other objects will be blurred. The mobile mapping community wants to avoid this at all costs, because most data is used to derive GIS systems, which need to be as accurate and complete as possible.

In this paper, we propose an effective post-filtering step using scene-specific constraints, by setting a sloppy detection certainty threshold, avoiding false negative detections (missed pedestrians), but additionally ensuring the removal of false positive detections using several effective post-processing steps. Furthermore we expand the system with additional small color based classifiers able to remove even more false positives. Finally we provide an elegant soft blurring approach for safeguarding the privacy of pedestrians inside the mobile mapping images.

The remainder of this paper is structured as follows. Section 2 presents related research, while section 3 discusses the data collection. This is followed by section 4 in which the proposed approach is discussed in detail. Finally section 5 elaborates on the obtained results while section 6 sums up conclusions and possible future improvements.

## 2 RELATED WORK

Pedestrian detectors come in different types and flavors. The main difference lies in the flexibility of the model, where we distinct between rigid and non-rigid approaches. Rigid approaches focus on an object always being in the same constellation, with only one large part trained as a model. Such approach is suggested by (Dalal and Triggs, 2005; Viola and Jones, 2001), where a rigid model based on gradients is fed to a support vector machine or a boosting step. A downside is that they are trained on a fixed frontal view of the object. Non-rigid approaches on the other hand try to model objects as a combination of deformable parts, existing of several rigid parts (arms, head, torso and legs), and a deformation relationship between them (Felzenszwalb et al., 2008). As pedestrians tend to move and change position frequently, we decided to use a non-rigid detector. Most pedestrian detectors discard color information, because of the wide variation in clothes and appearance. However more recent techniques like (Dollár et al., 2009; Dollár et al., 2010) show that including color information can have a significant increase in performance.

(Van Beeck et al., 2012) introduces a warping window approach where fast real-time vision-based pedestrian detection is obtained by calibrating the height and orientation of the pedestrian at each spe-

cific image location. We prove that we can apply a similar technique, as a post-processing step after the detection phase, by learning a relation between the height and position of an average pedestrian from a limited set of application-specific annotations.

(Puttemans and Goedemé, 2013) proves that using application-specific information, is one way to improve the accuracy of object detection algorithms. Similar rules apply for pedestrian detection, as far as the application allows you to find some application-specific constraints. In our application we exploit the fact that the camera is mounted on top of car, at a fixed position with respect to the ground plane, resulting in a relation between the position and the height of any given pedestrian. Furthermore we exploit the annotated training data to learn regions of interest, avoiding the processing of undesired image regions, like the sky or on top of buildings. (Cho et al., 2012; Peng et al., 2015; Dibra et al., 2015) describes a similar use of a ground plane assumption for 3D modeling and multiple camera view processing.

For privacy masking, several solutions have been proposed. (Tanaka et al., 2015) tries to define how much blurring is needed to reach a certain level of privacy. (Panagiotis, 2015) applies a simple block based blurring, whereas (Nakashima et al., 2015) suggests to use image melding, replacing a person’s face with a fixed neutral expression instead of blurring. Our application still demands masking, but to avoid the hardness of block based blurring, we propose to use a smooth soft blurring approach.

## 3 DATASETS

This research is developed on top of two mobile mapping datasets, which are made publicly available<sup>1</sup>, to encourage further research in this area.

The first dataset is a series of mobile mapping cycloramic images with a resolution of  $4800 \times 2400$  pixels, captured using a LadyBug 1 camera setup, in a quiet and calm urban area in the Netherlands. The captured images give a full 360 degree view from the surroundings of the car at any given position. The camera itself is fixed and mounted on the top of the roof of the car. The set has 450 images under daylight conditions. The dataset is used to develop and fine-tune the suggested approach.

The second dataset was captured using a LadyBug 2 camera, having a resolution of  $8000 \times 4000$  pixels, again mounted on top of the roof of the car, containing 45 images of a train and bus station in Belgium. We

<sup>1</sup><http://www.eavise.be/MobileMappingDataset>

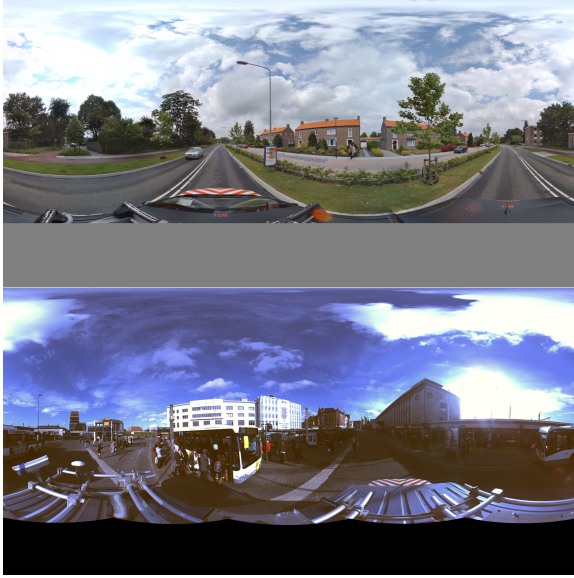


Figure 1: Example frames for both datasets used: (top) Dataset 1 - urban area in the Netherlands; (bottom) Dataset 2 - Belgian train and bus station.

used this dataset to prove that the developed approach is independent of the application-specific settings like camera setup and application environment, except for defining the actual height-position relation used to improve the detection success rate. An example of both mobile mapping datasets can be seen in Figure 1.

All database images were manually annotated to provide ground truth data for the actual locations of pedestrians. For the first dataset this led to 240 pedestrian annotations while the second dataset contained 1630 pedestrian annotations. The large difference is mainly due to the recorded surroundings, where a train and bus station is likely to have more pedestrians walking around in each mobile mapping image.

## 4 APPROACH

Our approach can be split into several processing blocks, as seen in Figure 2. First we create a limited amount of ground truth annotations for both datasets, needed for both building the height-position relation and inferring the color-specific constraints for learning the false positive elimination classifiers. The annotations are also used for validating each additional post-processing step. At runtime, we apply a multi-scale pedestrian detection algorithm on the input data provided in a sliding window manner, image, storing the detection results and their detection certainty score. Based on the annotations we apply a valid region reduction, a height-position location relation and

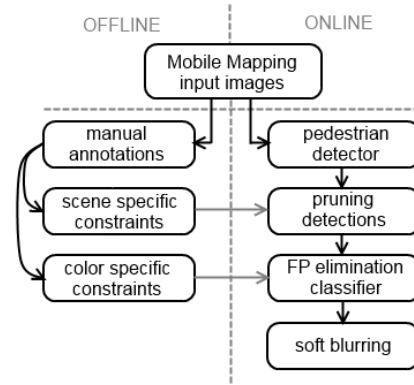


Figure 2: Block diagram of the suggested approach.

a certainty score thresholding, all leading to an efficient pruning of the obtained detections. For specific object classes that still trigger false positive detections, we design specific color based false positive elimination classifiers, which in turn further improve the results. The main goal is thus to remove as much false positives as possible and increase the resulting detection accuracy. The detections found are then passed on to an elegant soft blurring step to ensure privacy safeguarding.

For our research we used an implementation (De Smedt et al., 2012) of the cascaded Felzenszwalb latentSVM4 implementation, which uses a part based object detector for efficient pedestrian detection. The reason for this is quite straightforward. A part based detector is non-rigid and thus captures the different poses of pedestrians efficiently. On the other hand we also have a fast and optimized C++ implementation available. However, our post-processing is independent of the pedestrian detector used, so basically it could be replaced by any out-of-the-box pedestrian detector. This is one of the major benefits of our system, encouraging cross-dataset evaluation.

A downside to every pedestrian detection algorithm is that one must filter the output based on the object detection certainty score, by selecting a specific threshold and thus locking down on a specific point on the precision recall curve of that detector. If we decide to put the threshold too sloppy, we get an increase in false positive detections, while in the same time, reducing the amount of false negative detections (every single pedestrian will likely be returned). This would lead to an enormous amount of privacy masking, however also removing a lot of useful information from the image. This is unacceptable for the mobile mapping community, where the data is used to create high quality 2D and 3D GIS systems. On the other hand, if we put the threshold very strict, the amount of false positives will decrease drastically,

but we will get an increase in false negative detections, which in our application of privacy masking would not be acceptable. Our approach therefore uses a sloppy threshold for obtaining every possible pedestrian as a true positive detection, then subsequently using smart post-processing to efficiently remove as much false positive detections as possible.

## 4.1 Scale-space Location Relation

When considering our application of mobile mapping using a fixed 360 degree cycloramic camera, we know that the actual height position of the camera, compared to the environment, will be fixed, only if we assume a flat ground plane, and if that ground plane will never change drastically. This is a crucial scene constraint, allowing us to take into account that every pedestrian in the image, walking on the street or on the sidewalks, will have an average fixed height in relation to the position in the final cycloramic images. People closer to the car and thus to the camera will be larger, while people further away will move towards the camera's vantage points and thus be smaller. For any given horizontal line in the image, we can state that all pedestrians on that line will have the same average height, of course keeping in mind that we have a natural height variance within pedestrians.

### 4.1.1 Mapping Ground Truth Annotations

In order to model a height-position relation we started by mapping out the ground truth annotations collected on the first dataset. The effort of annotating a smaller part of application-specific data, to be used for deriving scene constraints, is small compared to training a complete new pedestrian detector (which needs much more annotations and processing time). The result of these manual annotations can be seen in Figure 4(a), where the height of each annotation is mapped in relation to the position, defined as the center of gravity. During the annotation phase only pedestrians on sidewalks, parks and roads were annotated. If a pedes-



Figure 3: Applying borders for minimal and maximal pedestrian height, as defined by the blue borders in 4(a).

trian would be standing on a balcony of a building, this pedestrian was not taken into account.

### 4.1.2 Model Fitting and Region Reduction

In relation to the data mapping seen in Figure 4(a) we fit a linear model to the data points and apply a search for image region boundaries. The red curve is the fitted linear relation to the mapped annotation data, which models the relation between pedestrian height and pedestrian position, relative to the camera position. The green borders are based on the assumption that we have a Gaussian data distribution compared to the fitted model, and that these borders should capture 99.8% of all detections using the rule of  $[-3\sigma, +3\sigma]$ . The reasons for this allowed model deviation are quite straightforward. First of all we have a natural deviation in pedestrian height, while secondly, due to the cars suspension, the camera height is not completely fixed to the ground plane. Thirdly, there is a possible deviation from the flat ground plane assumption caused by height differences due to sidewalks, defects in the road, speed bumps, etc. The blue borders define allowed position regions for pedestrians in the image, assuming the training data covers a wide variance of available pedestrian specific to the application. This is visualized in Figure 3 and allows us to immediately ignore detections that are outside these regions, removing about 50% of the image, and thus lowering the chance of false positive detections occurring.

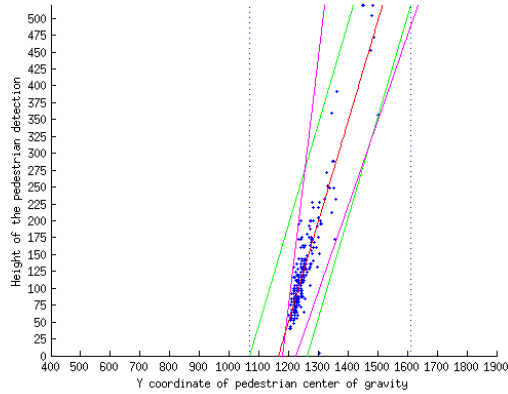
### 4.1.3 Applying Constraints on Detection Data

Figure 4(b) visualizes the detections obtained by our pedestrian detection algorithm. When applying the realistic pedestrian occurring boundaries, calculated from the annotated data in the previous subsection, we obtain the green dots, representing pedestrian detections in reasonable and allowed positions in the image. We do notice that this allows us to drop a significant amount of false positive detections. Subsequently we force the green borders on top of the green data, demanding that our detections also fit our height-position relation created from the manually annotated data. This in return removes a large part of the false positive detections, keeping only the red detections as acceptable pedestrian detections.

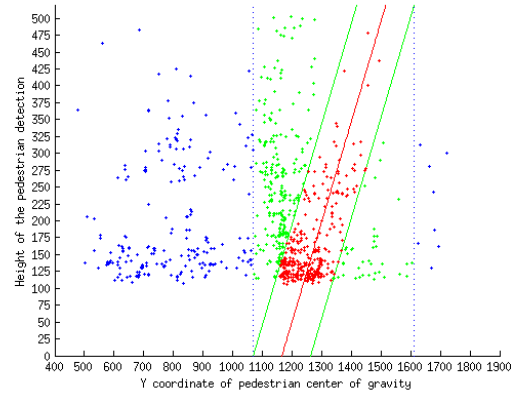
### 4.1.4 Updating the Distribution Constraint

We acknowledge that assuming a Gaussian distribution around the fitted linear height-position relation might not always be the best choice, especially when you consider the fact that when moving further from the car, differences in pedestrian height become less

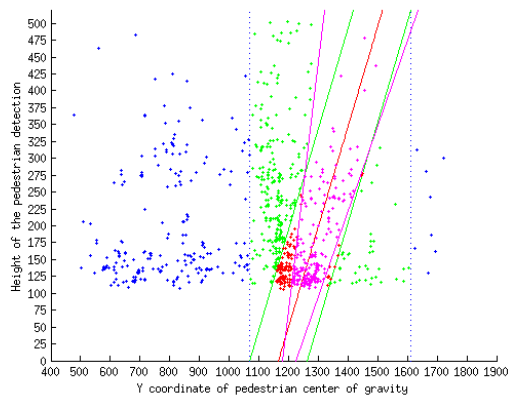




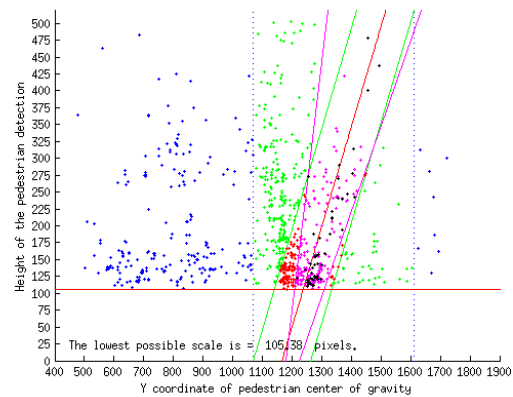
(a) Ground truth annotations with general and narrow bounds



(b) Detections with pruning steps applied



(c) Applying narrower bounds



(d) Score Threshold and smallest detection size

Figure 4: The height-position relation building process

obvious to notice, certainly at pixel level, whereas close to the car height differences are clearly visible. Therefore we updated the green borders, to closely map the correct distribution of the annotation data, which can be seen in Figure 4(a) as the magenta borders. Applying those updated constraints on the actual detection output, again removes several false positive detections, resulting in the magenta colored detections seen in Figure 4(c).

#### 4.1.5 Detection Certainty Thresholding

Before applying all the constraints defined in the previous subsections on top of the detection output, we decided to put the detection certainty threshold very sloppy, to ensure that the amount of false negative detections is close to 0%. Now that we have automatically removed multiple false positive detections, we can look back to this setting and adapt it to our application-specific needs. Due to less cluttered images filled with detections, since most false positives

are removed now, it becomes easier to select a decent score threshold for our application. From experience in using pedestrian detectors in the wild, we learned that the used LatentSVM4 detector almost never returns valid pedestrian detections when the certainty score is below 0. Of course this value is application specific and can change drastically when considering other application fields. In our application, detections with lower scores mainly resemble objects that have similar feature descriptions, like a smaller tree or a traffic sign, but in 99% of the cases, they do not match actual pedestrians. Since we want to avoid blurring too much valuable image information, we enforce an extra pruning rule, demanding a detection certainty score equal or above 0. This results in the black detections, seen in Figure 4(d).

#### 4.1.6 Visually Verifying the Filtered Detections

When visually checking the data, we wondered why very small pedestrians in the background where ig-

nored by the pedestrian detection interface. As seen in Figure 4(d) we calculated the smallest retrieved detection height by the DPM detector, which had a height of 105 pixels. Considering this in relation to the pre-trained pedestrian model, this is actually normal, because the model is trained with a fixed training sample height of 124 pixels, keeping a small area of background around the pedestrian, also called padding. At detection time, the model’s dimensions always limit the smallest possible detection height, so if we would like to include these smaller pedestrians, we should first upscale the input data. However we should keep in mind that this introduces image artifacts which could interfere with the pedestrian detector. In our application this is no problem, since pedestrians with a height smaller than 100 pixels are already privacy secure and impossible to recognize when looking at the complete mobile mapping image of  $8000 \times 4000$  pixels (Tanaka et al., 2015).

#### 4.2 Color-Based Removal of Pedestrian-Like Detections

Even with all the proposed post-filtering steps applied, we noticed that some object classes continuously succeeded in triggering false positive detections. Take for example the case of small traffic signs indicating the traffic flow when entering a roundabout, as seen in Figure 5. As humans we clearly see the difference between a pedestrian and this rigid traffic sign. However due to the specific nature of pedestrian detection algorithms, it is normal that these false positive detections occur. First of all, the used algorithm (Felzenszwalb et al., 2008) ignores color information, since the variety of color in pedestrians is enormous. Secondly as feature it uses edge information of deformable parts. And this is exactly where the biggest problems occur. The mentioned traffic sign has a top part that is very similar to a head and a middle and bottom part that have similar feature responses as a human body. Since the body and the head are parts with a big weight in part-based pedestrian models, it is important to add an extra pruning step to remove these false positive detections that are still



Figure 5: Example of the need of an extra filter for traffic signs still passing the post-processing steps.



Figure 6: Positive, negative training and test set for traffic sign filtering.

classified as valid detections by our pipeline. Especially in the context of mobile mapping it is important that crucial road information is not filtered or blurred out due to privacy reasons, because many clients interested in this data are looking for exact locations of traffic signs like this, e.g. to keep an automated index of road sign conditions.

To avoid these kind of problems we propose a simple pruning step using a small Naive Bayes classifier. This machine learning technique takes a limited set of positive and negative training samples and, based on some very simple color-based features calculated from the training data, decides whether a valid detection should still be classified as pedestrian or not. We prefer using a machine learning approach towards setting hard thresholds on basic features, because it is more robust in finding the optimal separation between classes once more training data is supplied.

As seen in Figure 6, we use a small positive and negative training set (both containing only 5 samples), where we tried to include as much traffic sign like pedestrians in the negative set as possible (by looking for matching colors), to avoid that those would now get filtered out, e.g. when someone is wearing a bright jacket. Finally we constructed a small test set to evaluate the success rate of our classifier.

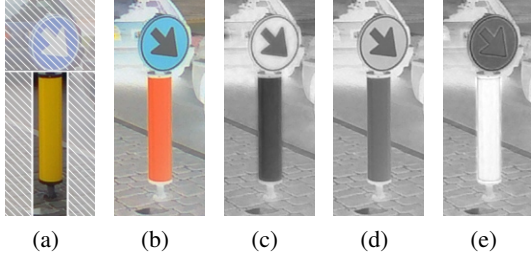


Figure 7: Naive Bayes Features: (a) original (b) CMY(K) (c) C response (d) M response (e) Y response.

From each training sample a set of simple visual features are calculated. In this case the most distinct feature is the bright yellow color of the ‘body’ part of the traffic sign. We separate the top 30% of the image and then split up the bottom 70% in 3 equal regions, as seen in Figure 7(a). middle area is then transferred to the CMYK color space (Figure 7(b)) because the traffic sign has a very low response in the C layer (Figure 7(c)), an average response in the M layer (Figure 7(d)) and a high response in the Y layer (Figure 7(e)). This behavior is not equal for pedestrians. We take the average CMY values for this smaller window and use that as feature vector for each positive and negative sample. The K channel is simply ignored.

Finally when running the classifier on the test set provided, the samples were all classified correctly either as pedestrian or as traffic sign and thus the simple classifier proved to work as an effective post-filtering step. Similar behavior was detected for specific kind of bushes, again in this case, an extra small Naive Bayes filter could be constructed. The advantage of this approach is that at post-processing time, the calculation of these extra filters is computationally very cheap ( 1ms) due to the very simple features used and thus a small cost for a better classification result.

### 4.3 Soft Blurring Approach

The final step of our proposed pipeline is to obtain the valid detected pedestrian regions and apply a local apply a privacy-safeguarding filter to them. In our cooperation with mobile mapping companies it became clear that they want to manually define which part of the detection is being blurred. Therefore we provided the option for both pedestrian and face region blurring. An intuitive way to apply privacy-safeguarding would be to apply a standard Gaussian blurring filter. One of the main downsides to this is the existence of very prominent edge artifacts which cannot be removed, as seen in the left part of Figure 8. We would prefer a blurring filter that is not as strong on the edges, as seen in the right part of Figure 8, but

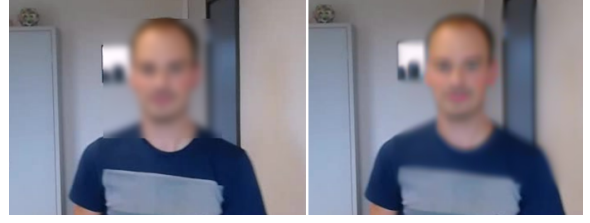


Figure 8: Blurring filters (left) standard Gaussian blur (right) smooth blurring filter.

which is strong in the middle and softens up towards the edges of a detection. This ensures privacy but the end result is visually more pleasing.

Instead of convolving the image region with a Gaussian kernel with a fixed size and sigma, we propose a convolution with an adaptable Gaussian kernel, where the sigma ( $\sigma_{kernel}$ ) is defined as a function of the normalized pixel distance  $\Psi$  to the center of the detection itself as described in equations (1), (2) and (3). To ensure that the blurring is proportional for differently sized pedestrian detections, we add an extra size dependency  $\Delta$ , which takes into account the area of the detection found compared to the area of the original image. This ensures that in the end each detection is equally blurred.

$$\Psi = 1 - \frac{d(center_{detection}, position)}{r} \quad (1)$$

$$\Delta = \frac{area(pedestrian)}{area(image)} \quad (2)$$

$$\sigma_{kernel} = 0.1 + (\Delta \Psi^2) \quad (3)$$

We apply this soft blurring filter to every pedestrian detection in a given input image, blur out the detected pedestrians or their associated face region and make the captured mobile mapping image privacy safe. In our application we applied face blurring which can be seen in Figure 10 and 9. This is simply

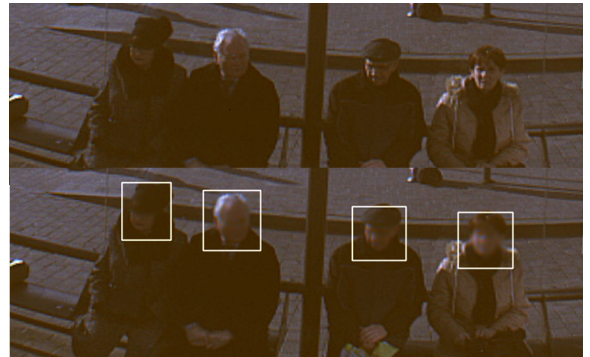


Figure 9: Close-up of privacy smoothing using only the face region of the detection.





Figure 10: Applying soft blurring on filtered pedestrian detections, but limiting the blurring to face regions only.

passed as an extra parameter to our smooth blurring function. In order to make the blurring regions more visible we also visualized the actual detections.

## 5 RESULTS

We applied the same post-processing steps discussed in the previous sections (enforcing valid pedestrian regions, applying a height-position relation and adding a scoring threshold) to the second dataset and got similarly good improvements. Only the color-based Naive Bayes classification was left out, since the specific object class (roundabout traffic sign) did not occur inside the second dataset. We did not explicitly look into an object class specific Naive Bayes filter for the second dataset, but if such false positive triggering object class would occur, one could simply train a classifier for that class using our software. The result of pruning the detection output can be seen in Figure 11, while visual results of applying these constraints

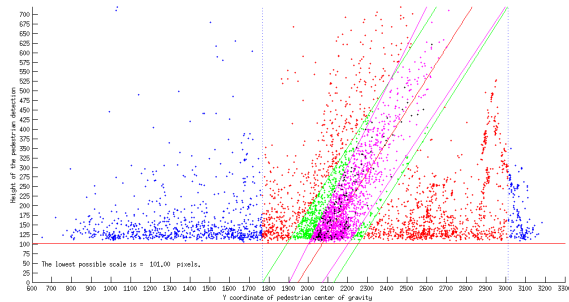


Figure 11: Applying all post-processing steps to the output of the LatentSVM 4 INRIA based pedestrian detector applied on dataset 2.

can be seen in Figure 13. Especially pay attention to the false positive detections on the car that are disappearing as well as some of the double detections.

In order to make sure that we actually achieved an improvement over simply using the out-of-the-box available pedestrian detection algorithm, we evaluated the number of true positive, false positive and false negative detections after applying the different post-processing steps discussed in section 4. The result of this comparison can be seen in Table 1. Subsequently, using precision-recall curves, we visualized the accuracy gained by applying the mentioned post-processing steps to the detection results on dataset 2, which can be seen in Figure 12. Notice that an out-of-the-box object detector already experiences a large accuracy drop when doing cross-dataset evaluation, and that there is a substantial accuracy gain when applying our post-processing steps. Since the

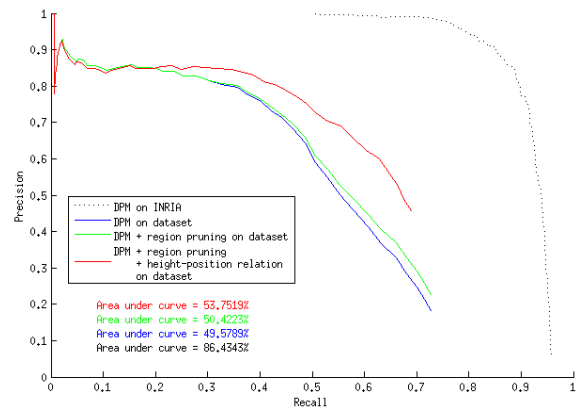


Figure 12: Precision-Recall curves generated for dataset 2 with all post-processing steps applied and the reported accuracy using the area under the curve measurement.

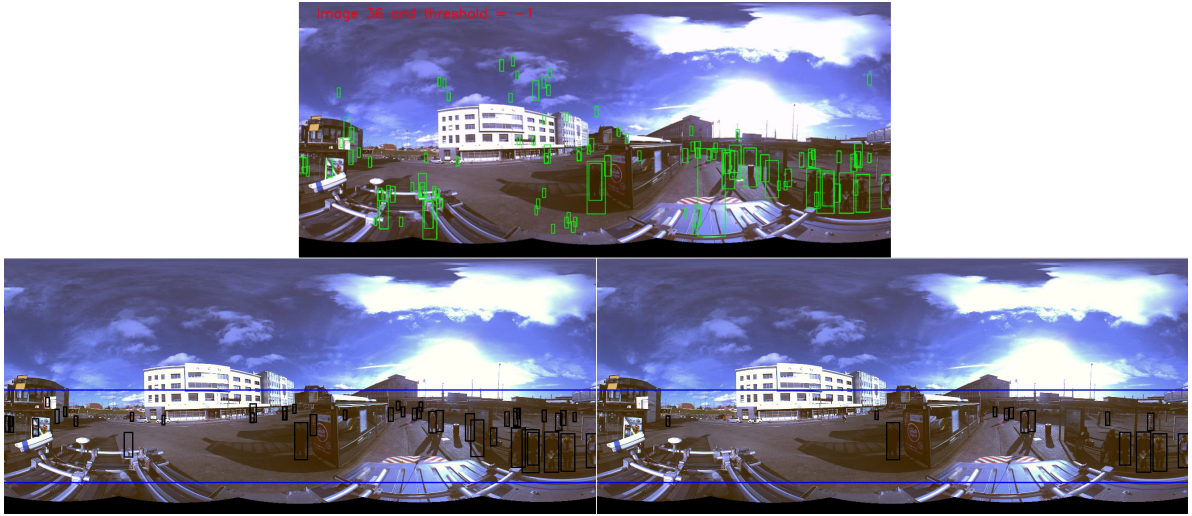


Figure 13: Example of applying post-processing steps to data from the second dataset. (top) original detections at score threshold -1; (bottom left) after pruning; (bottom right) after height-position relationship and score  $> 0$ .

detection analysis for dataset 2, shown in Figure 11, proves that the minimum object size found by the used DPM model is 101 pixels, we ignored possible ground truth annotations on pedestrians smaller than 100 pixels, to make an as accurate precision-recall curve as possible. The remaining false negative detections are mainly due to people sitting on benches, riding bikes or motorcycles, which are less likely to get detected by the used pedestrian DPM model and which is a known issue.

We do acknowledge that this solution is far from 100% fail prove. There are still some bottlenecks that should be taken into account. The overall approach is generated to improve the output of any available pedestrian detector without retraining the actual DPM model specific to the application. However, up till now there is not yet a single of-the-shelf pedestrian detector which is able to detect every single pedestrian out there in any given application, especially when performing cross-dataset validation (Torralba et al., 2011). While our approach focuses mainly on improving the recall rate of our detector, as seen in Figure 12, getting the precision of pedestrian detectors to 100% in any given application is still a very

Table 1: Comparison of TP, FP and FN values after each post-processing step for first dataset. To obtain a clear benefit of applying these techniques, we ran the original DPM detector at a score threshold of -1 like in the visual results shown in Figure 13.

	#TP	#FP	#FN
<b>DPM orig.</b>	928	4159	349
<b>After pruning</b>	928	3182	349
<b>After height-position</b>	852	1015	384

challenging task and an actively researched topic.

During this research we made a visualization showing the influence of changing the threshold on the detection certainty, from a very sloppy value to a very strict value, in relation to the amount of false positive detections produced. This clearly shows the influence of changing this parameter in search of an ideal setting for any given application. The video can be found at: <https://youtu.be/-xrBg8sDDOQ>.

## 6 CONCLUSION AND FUTURE WORK

The goal of this paper is to efficiently blur pedestrians in mobile mapping images to avoid privacy related issues while safeguarding as much image information as possible. By using an off-the-shelf pedestrian detector trained on a different dataset and setting a sloppy confidence threshold, we proved that applying efficient post-processing filters, based on application-specific constraints, e.g. a height-position relation, can greatly improve the detection outcome. In addition to the proposed height-position filtering step, we supply additional easy to train lightweight Naive Bayes filters for objects that still trigger false positive detections, e.g. roundabout traffic signs, without the need of large amounts of annotated training data.

We prove that in a specific situation, we can use pre-trained pedestrian detection models, but, given a limited amount of manual annotations on a situation specific dataset, we can boost the detection accuracy enormously by exploiting scene-specific constraints, e.g a known ground plane assumption. Fi-



nally we proposed an efficient soft blurring alternative to a standard Gaussian blurring filter, for privacy masking reasons, by adaptively changing the parameters of the Gaussian kernel used for the convolution with the found pedestrian detections.

Since the processing of mobile mapping images is being done off-line, and time and resource management was not the focus of this research, we do not need to concern about running the detector on every image location, which is computationally quite expensive. One could argue that running the out-of-the-box object detector multi-scale on every image position is actually a waste of resources and computing time. As future work we suggest to integrate our post-processing steps inside the actual pedestrian detection algorithm, enormously reducing the processing time needed for a single mobile mapping image. This might open up the possibility to do the processing on-line, while capturing the actual data. This would be better for industrial partners, since privacy issues would be solved completely, due to the privacy sensitive data not being physically stored anymore.

Our application focuses on detecting pedestrians walking on the modeled ground plane, which raises a new problem. People standing on a balcony, sitting on a bench, lying on the grass or driving a bike, will not fit into this ground plane assumption and will thus simply be filtered out by our approach. We could improve our approach by using multiple detection models, for these different pedestrian classes and then apply separate post-filtering rules for each detector.

One could not disagree that even with the current bottlenecks, that this work is valuable for people handling privacy sensitive mobile mapping data. This research allows users to automatically remove privacy sensitive data from their captured datasets, without the need of manually handling each image (which would be very costly and time consuming). It allows users to grab off-the-shelf available pedestrian detectors, add them to the system, and use a limited manual input in their application field to derive the post-processing rules. This highly benefits the companies because they do not need to put huge amounts of time and resources into building an application-specific pedestrian detector themselves, needing thousands of pedestrians to be manually annotated.

## ACKNOWLEDGMENTS

This work is supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT) via the IWT-TETRA project TOBCAT and via the IWT-TETRA project RaPiDo. We would also like to thank Vansteelandt BVBA

and Grontmij Belgium, the companies who provided the cycloramic image datasets during these projects, which were used to develop and test this approach.

## REFERENCES

- Cho, H., Rybski, P. E., Bar-Hillel, A., and Zhang, W. (2012). Real-time pedestrian detection with deformable part models. In *IVS*, pages 1035–1042. IEEE.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893. IEEE.
- De Smedt, F., Struyf, L., Beckers, S., Vennekens, J., De Samblanx, G., and Goedemé, T. (2012). Is the game worth the candle? Evaluation of OpenCL for object detection algorithm optimization. *PECCS*, pages 284–291.
- Dibra, E., Maye, J., Diamanti, O., Siegwart, R., and Beardsley, P. (2015). Extending the performance of human classifiers using a viewpoint specific approach. In *WACV*, pages 765–772. IEEE.
- Dollár, P., Belongie, S., and Perona, P. (2010). The fastest pedestrian detector in the west. In *BMVC*, volume 2, page 7. Citeseer.
- Dollár, P., Tu, Z., Perona, P., and Belongie, S. (2009). Integral channel features. In *BMVC*, volume 2, page 5.
- Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8. IEEE.
- Nakashima, Y., Koyama, T., Yokoya, N., and Babaguchi, N. (2015). Facial expression preserving privacy protection using image melding. In *ICME*, pages 1–6. IEEE.
- Panagiotis, I. (2015). Preventing privacy leakage from photos in social networks. In *CCS2015*. ACM.
- Peng, P., Tian, Y., Wang, Y., Li, J., and Huang, T. (2015). Robust multiple cameras pedestrian detection with multi-view bayesian network. *Pattern Recognition*, 48(5):1760–1772.
- Puttemans, S. and Goedemé, T. (2013). How to exploit scene constraints to improve object categorization algorithms for industrial applications. In *VISAPP*, volume 1, pages 827–830.
- Tanaka, Y., Kodate, A., Ichifuji, Y., and Sonehara, N. (2015). Relationship between willingness to share photos and preferred level of photo blurring for privacy protection. In *ASE BigData & SocialInformatics*, page 33. ACM.
- Torralba, A., Efros, A., et al. (2011). Unbiased look at dataset bias. In *CVPR*, pages 1521–1528. IEEE.
- Van Beeck, K., Goedemé, T., and Tuytelaars, T. (2012). A warping window approach to real-time vision-based pedestrian detection in a truck’s blind spot zone. In *ICINCO*, volume 2, pages 561–568.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*, pages 1–511.